University of Ljubljana
Faculty of Electrical Engineering
Laboratory of Photovoltaics and Optoelectronics

CROWM

Combined ray optics / wave optics model

User Manual

Version 2.6

Ljubljana, May 2014

Laboratory of Photovoltaics and Optoelectronics
Faculty of Electrical Engineering
University of Ljubljana
Tržaška 25, SI-1000 Ljubljana
Slovenia
http://lpvo.fe.uni-lj.si

Head of LPVO:

Prof. Dr. Marko Topič
E-mail: marko.topic@fe.uni-lj.si
Tel.: +386 1 4768 470
Fax: +386 1 4264 630

Author and contact person of CROWM:

Dr. Benjamin Lipovšek
E-mail: benjamin.lipovsek@fe.uni-lj.si
Tel.: +386 1 4768 845

Official CROWM webpage:

http://lpvo.fe.uni-lj.si/en/software/crowm/

Contents

1. INTRODUCTION 1

1.1 Short overview 1

1.2 Description of the general simulated structure 2

1.3 Description of the optical model 3

2. INSTALLATION OF CROWM 5

2.1 Installation of CROWM program files 5

2.2 Installation of the hardware security key 5

2.3 Installation of the Matlab Compiler Runtime (MCR) 6

2.4 Files and folder structure of CROWM 6

3. RUNNING CROWM FROM COMMAND LINE 7

3.1 Execution of a single simulation 7

3.2 Running a batch of simulations 8

3.3 Running CROWM from other programs 8

4. RUNNING CROWM USING THE GRAPHICAL USER INTERFACE (GUI) 9

4.1 Main simulation panel 9

4.2 Input file editor 14

4.3 Texture viewer 20

4.4 Device structure viewer 21

4.5 Ray propagation viewer 22

4.6 Results viewer 23

5. DESCRIPTION OF THE INPUT FILES 25

5.1 Simulation input files (".txt", located in "In" folder)..... 25

5.2 Complex refractive index files (".nk", located in "nk" folder)..... 27

5.3 Spectrum files (".spc", located in "Spectrum" folder)..... 27

5.4 Texture files (".txt", located in "Texture" folder)..... 28

5.5 Simulation batch files (".bat", located in root folder)..... 29

6. DESCRIPTION OF THE OUTPUT FILES 31

6.1 Log file ("_Log.txt")..... 31

6.2 Reflectance, transmittance, absorptance files ("_RTA.txt")..... 31

6.3 Short-circuit current density file ("_Jsc.txt")..... 32

6.4 Ray propagation figure file ("_Rays.fig")..... 32

1. INTRODUCTION

1.1 Short overview

The optical simulator CROWM (Combined Ray Optics / Wave Optics Model) has been developed for optical simulation of multi-layered optoelectronic structures (e.g. thin-film solar cells) comprised of a single thick (incoherent) layer and an arbitrary number of thin (coherent) layers located on either side of the thick layer. The thick layer can be either flat or textured; in the latter case, arbitrary 1D or 2D textures with lateral and vertical features in the range of micrometres or millimetres can be applied on either side of the layer (see section 1.2 for a detailed description of the general device structure that can be simulated by CROWM).

The main input parameters of a typical CROWM simulation are the illumination conditions, the exact structure of the simulated device, and the optical properties of the materials used in the device. The main output parameters are the total reflectance and the total transmittance of the entire device, and the absorptances within each of the individual layers (see chapters 5 and 6 for a detailed description of the input and output files).

CROWM is equipped with a fully-featured graphical user interface (GUI) which can be used for running the simulations, for viewing and editing the input files, for drawing the 3D schematics of the textures and entire device structures, and for the analysis and comparison of simulation results (see chapter 4 for a detailed description of the GUI features). Optionally, CROWM can also be run directly from the command line or even from within other programs such as Matlab (see chapter 3).

The purpose of this manual is to inform the user about the various features and settings of CROWM and at the same time explain how to best utilise the software in order to perform accurate and reliable simulations. The user is expected to be familiar with the general topics in optics, optoelectronics and photovoltaics.

1.2 Description of the general simulated structure

The schematic of the general optoelectronic device structure that can be simulated by means of the optical simulator CROWM is presented in Figure 1.1. The device consists of:

- A single **thick incoherent layer** with the thickness in the range of 10 μm – 10 mm.
- Optional one-dimensional or two-dimensional **surface textures** applied to the top surface and/or the bottom surface of the thick incoherent layer. The textures at the top and bottom surface do not need to be the same. The lateral and vertical features of the textures are in the range of micrometres or larger. The textures are assumed to be **periodic in both directions** (see section 5.4) and should be continuous and single-valued everywhere (i.e. a single *z* value per (*x*,*y*) coordinate).
- An arbitrary number of **thin films** deposited on either (or both) side of the thick incoherent layer. The thicknesses of thin films should be in the range of 1 nm – 1 μm and are assumed to be much smaller than the lateral and vertical features of the textures. As a result, the thin films are assumed to be locally flat (parallel to each other and the surface of the thick incoherent layer). As an option, **one** of the thin films can be treated **incoherently**.
- The **incident medium** (typically air) and the **medium in transmission**, which are assumed to be infinitely thick.

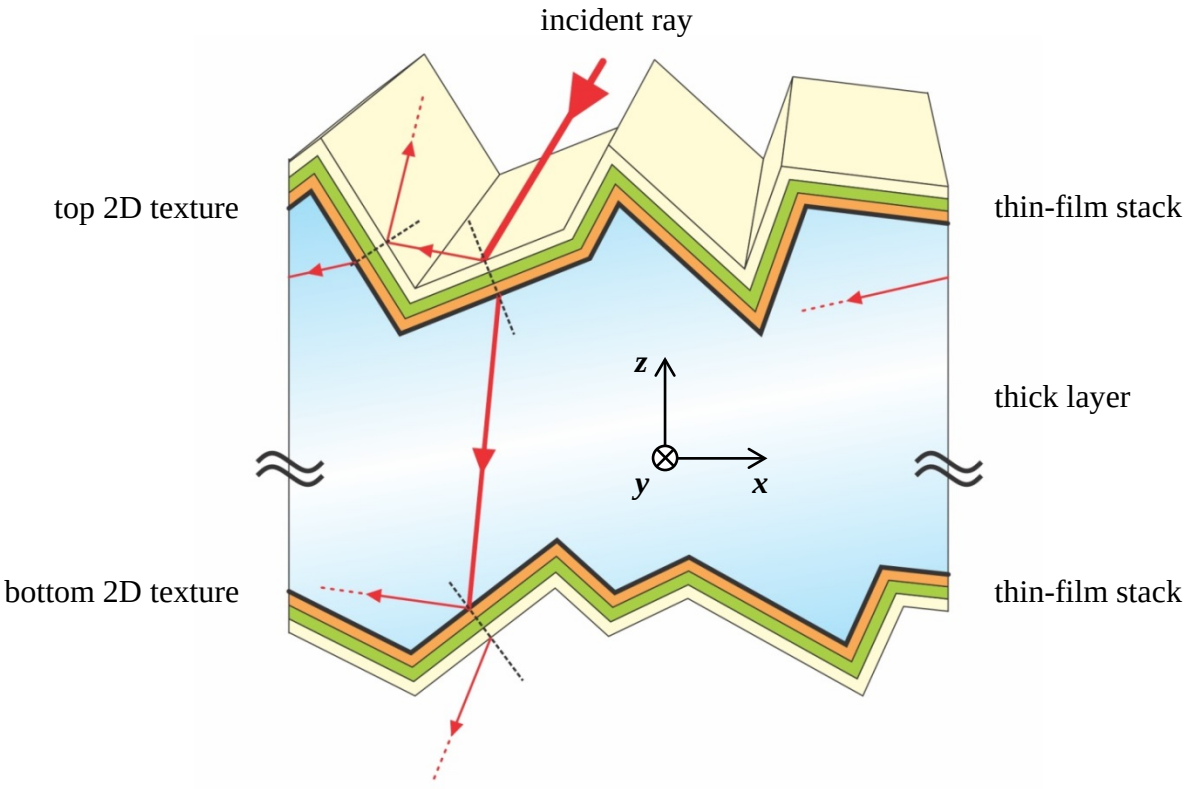


Figure 1.1: Schematic of the general device structure that can be simulated in CROWM.

1.3 Description of the optical model

Since the thickness of the thick layer is much larger than the effective wavelength of light, **incoherent light propagation** can be assumed within this layer. In the thin-film stacks, on the other hand, layer thicknesses in the range of the effective wavelength of light result in fully **coherent light propagation**. In order to simulate this combined incoherent/coherent light propagation, CROWM combines two numerical approaches for the optical analysis of the thick and thin components of the simulated structure:

- Light propagation through the incident medium, the thick surface-textured layer, and the medium in transmission is analysed by means of **incoherent three-dimensional ray tracing** based on geometric optics. The incident illumination, which can be applied under an arbitrary incident angle (combination of azimuth and zenith angles), is divided into a number of rays which are then traced through the structure. The intensities and the directions of the rays are determined by the texture morphology (refraction and reflection effects at the interfaces), the optical absorption in the material (in the case of absorptive layers), and the calculated reflectances and transmittances through the thin-film components (calculated with a different model; see below). The textures are assumed to be periodic, in which case the boundary conditions are applied to the lateral borders of the ray tracing domain (i.e. a ray reaching the left border will reappear at the right border, and vice-versa, as schematically shown in Figure 1.1).
- The thin-film stacks in the simulated structure are treated by means of **fully-coherent transfer matrix formalism**, which is a widely established one-dimensional method for simulation of light propagation through a stack of thin layers, taking into account coherence effects such as destructive and constructive interference. The thin-film stacks are assumed to be parallel to each other (i.e. the textures of the thick layers are transferred to all the interfaces within the thin-film stacks, as schematically shown in Figure 1.1).

Since the total thicknesses of the thin-film stacks are assumed to be much smaller than the thickness and the texture features of the thick layer, the reflected and the transmitted rays through the thin-film stacks will always emerge from the same point (relative to the local surface) where the incident ray impinged upon the thin-film stack (there are no lateral shifts, as also shown in Figure 1.1). In other words, infinitely thin thin-film stacks are assumed when calculating the location of the reflected and transmitted rays in the incoherent components of the simulated structure.

2. INSTALLATION OF CROWM

2.1 Installation of CROWM program files

The optical simulator CROWM runs on Microsoft Windows XP / Vista / 7 operating systems. It is distributed as a single installation file **"CROWM_Installer.exe"** located on the installation medium. This installation package contains all the necessary program files and folders required to perform CROWM simulations. The simulator can be installed to an arbitrary folder on the hard drive, e.g. "D:\CROWM\". After installation, the folder structure within the main (root) program folder must not be changed in any way, although the main program folder itself (with all the sub-folders and files) can be freely copied elsewhere on the system. To uninstall CROWM from your operating system, simply delete the main program folder with all the contents.

2.2 Installation of the hardware security key

CROWM is copy-protected by means of a hardware security key, which needs to be connected to a free USB port during each simulation. Before starting to use CROWM, the drivers for the hardware security key need to be properly installed on the system.

The FTDI drivers for the hardware security key are located in the **"Drivers"** folder on the CROWM installation medium. The same drivers are used for all versions of Microsoft Windows operating system, however, the installation procedure differs from system to system. Therefore, to properly install the drivers, consult the appropriate installation guide located in the **"Documents"** folder on the CROWM installation medium (three guides are available for Microsoft Windows XP, Vista, and 7 operating systems).

During the installation procedure, when the installation wizard asks for the location of the drivers, simply point to the **"Drivers"** folder on the CROWM installation medium. Please note that these drivers differ significantly from the official drivers located at the FTDI webpage (www.ftdichip.com) – the official drivers will not work with the CROWM hardware security key.

If a communication error with the key should occur at any time during a simulation (the event is also logged in the log file; see section 6.1), try reconnecting the key and re-run the simulation.

2.3 Installation of the Matlab Compiler Runtime (MCR)

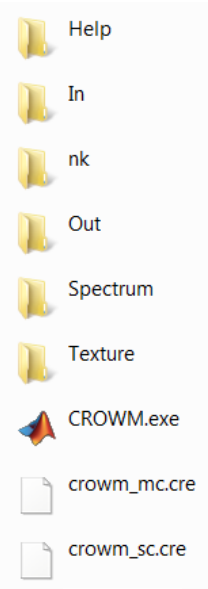
The optical simulator CROWM was written and compiled in the Matlab programming language. In order to run CROWM on operating systems which don't have Matlab installed, the Matlab Compiler Runtime (MCR) must be installed in order to provide all the necessary shared functions and libraries required by the simulator.[1]

The MCR installation file **"MCR_Installer.exe"** is located in the **"MCR"** folder on the CROWM installation medium. Before starting to use CROWM, MCR needs to be installed to an arbitrary folder on the system. After a successful installation, reboot of the system is also required.

Detailed instructions for installing the MCR libraries are located at the official Matlab support page: http://www.mathworks.com.au/help/compiler/working-with-the-mcr.html

2.4 Files and folder structure of CROWM

After a successful installation of CROWM, the following sub-folders and files can be located in the main program folder:



- **Help** – this folder contains the User Manual (this file)
- **In** – this folder contains the simulation input files (see section 5.1)
- **nk** – this folder contains the complex refractive index files for each layer of the simulated device (see section 5.2)
- **Out** – here is where the simulation results are stored (see chapter 6)
- **Spectrum** – this folder contains the spectrum files used in the simulations (see section 5.3)
- **Texture** – this folder contains the files with the description of the textures applied to the front and/or back surface of the thick incoherent layer (see section 5.4)
- **CROWM.exe** – CROWM GUI execution file (see chapter 4).
- **crowm_mc.cre** – simulation execution file for running CROWM simulations on multiple processor cores ("mc" = multi-core)
- **crowm_sc.cre** – simulation execution file for running CROWM simulations on a single processor core ("sc" = single-core)

[1] MCR release 2014a (8.3) 32-bit is required to run CROWM v2.6.7.

3. RUNNING CROWM FROM COMMAND LINE

3.1 Execution of a single simulation

When all the required input data files are present within the corresponding folders (see chapter 5 for a detailed description of the input files) and the hardware security key is connected to a free USB port, a CROWM simulation can be executed:

- directly from the **command line** (e.g. Command Prompt),
- by means of a **batch file**, or
- by using the **graphical user interface** (GUI), which is the preferred option.

When running CROWM from the command line, the execution command consists of the simulation execution file (either **"crowm_mc.cre"** for multi-core or **"crowm_sc.cre"** for single-core processing) followed by four arguments in order:

- **name of the input file** – e.g "Input.txt" (located in the "In" folder)
- **name of the top texture file** – e.g. "Texture_top.txt" (located in the "Texture" folder)
- **name of the bottom texture file** – e.g. "Texture_bot.txt" (located in the "Texture" folder)
- **name of the output files** – e.g. "Out_name"

This is an example command:

```
D:\CROWM\crowm_mc.cre Input.txt Texture_top.txt Texture_bot.txt Out_name
```

Please note that in the case of flat textures, the texture name in the execution command can be omitted and replaced with a single character **"0"** (zero). No texture input files are required for specifying a flat texture. Following is an example command for a simulation in which the front (top) surface of the thick incoherent layer is assumed to be flat:

```
D:\CROWM\crowm_mc.cre Input.txt 0 Texture_bot.txt Out_name
```

Once the simulation has been executed, the status of the simulation is displayed in the command window, and the simulation results are stored in the "Out" folder. The names of all the output files created during a simulation consist of the name specified in the command (e.g. "Out_name") followed by different endings (e.g. "Out_name_Jsc.txt" in the case of Jsc files; see chapter 6 for a detailed description of the output files).

3.2 Running a batch of simulations

Batch files are a convenient way for running a series of CROWM simulations. Batch files are simple text files with ".bat" extension which contain one or more commands that are sent directly to the command line after the execution of the file. Thus, extensive series of simulations can be prepared in advance (i.e. a list of commands in the batch file) which can then be executed at leisure (e.g. an overnight batch). As an example, CROWM package includes a batch file called **"Example_batch.bat"**. The file contains execution commands for three example simulations, all of which use the same cell structure, but different textures applied to the front and back surface of the thick incoherent layer. The file can be opened and edited in any text editor, such as Notepad. The contents of the file are shown in Figure 3.1.

```
echo off
echo ************************************************************
echo *              CROWM v2.6 - SIMULATION BATCH FILE            *
echo ************************************************************
echo An example batch file for a series of three CROWM simulations
echo of the same a-Si:H solar cell with different glass textures.
echo on

crowm_mc.cre a_SiH_Cell.txt 0              0 a_SiH_Cell_Flat
crowm_mc.cre a_SiH_Cell.txt Triangular.txt 0 a_SiH_Cell_Textured_top
crowm_mc.cre a_SiH_Cell.txt 0 Triangular.txt a_SiH_Cell_Textured_bot
```

Figure 3.1: Contents of the example batch file.

3.3 Running CROWM from other programs

The optical simulator CROWM can be run easily from within other applications such as Matlab, which makes it fairly simple to implement custom simulation procedures for automatic program execution and optimisation of the simulated structures.

In the case of Matlab, a CROWM simulation can be executed by means of the **"dos"** command, which sends the string in the argument directly to the command line, as shown by the following example:

```
dos('crowm_mc.cre Input.txt Texture_top.txt Texture_bot.txt Out_name')
```

4. RUNNING CROWM USING THE GRAPHICAL USER INTERFACE (GUI)

4.1 Main simulation panel

The main simulation panel, which is used to setup and execute CROWM simulations, is displayed when running the "**CROWM.exe**" file in the CROWM installation folder. From this panel, all the other features of the GUI can be readily accessed by clicking on the appropriate buttons, as described in the following. The screenshot of the main simulation panel is shown in Figure 4.1.

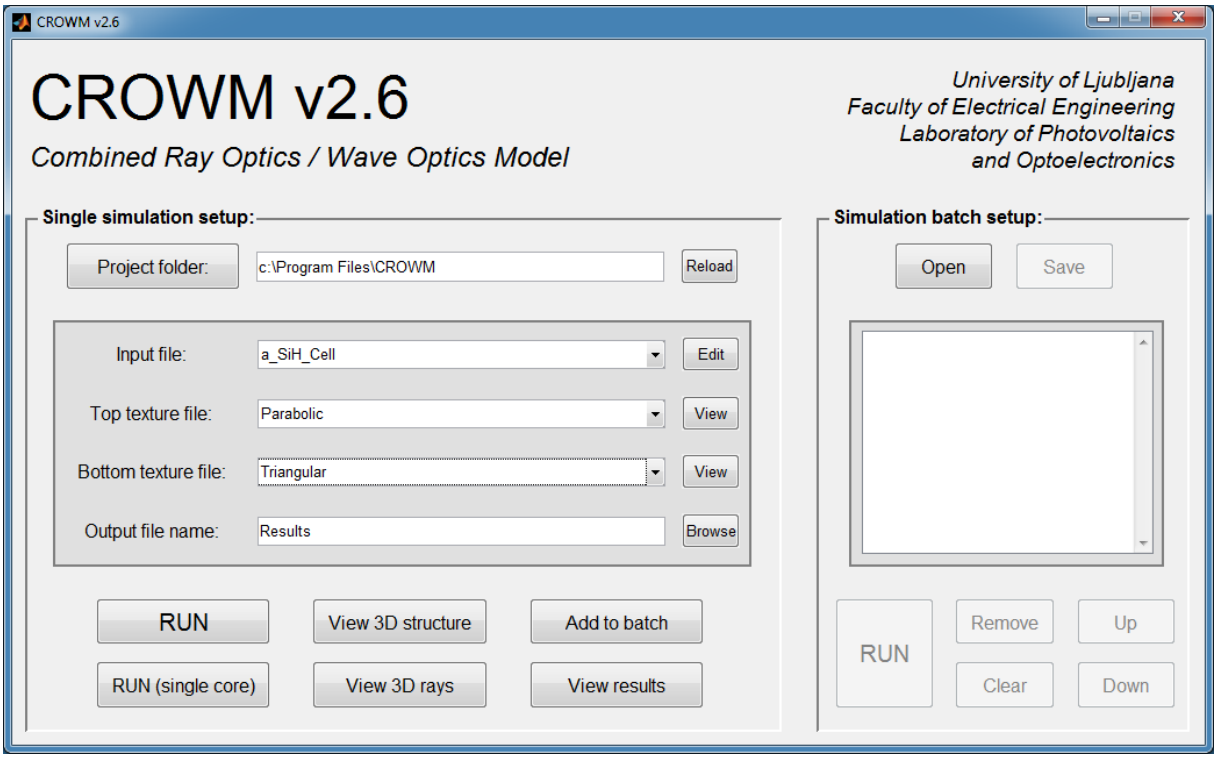


Figure 4.1: Main simulation panel.

The main simulation panel is divided into two sections. The first section is titled ***"Single simulation setup"*** and is used to setup a single CROWM simulation, whereas the second section titled ***"Simulation batch setup"*** is used to prepare a series of simulations (batch) which are to be run in succession.

4.1.1 Single simulation setup

This section is used to setup single CROWM simulations and also access the various other features of the GUI. The functionality of the individual controls is explained below.

4.1.1.1 Project folder *(button & text box)*

Select the current project folder. The project folder must contain all the folders required for running a CROWM simulation, as described in section 2.4. The user can either browse for the folder by clicking on the button, or enter the folder location directly into the text box. By default, the project folder is set to the CROWM installation folder.

4.1.1.2 Reload *(button)*

Reload all the input and texture files located in the "In" and "Texture" folders of the project folder and list them into the input and texture menus (see below). This is useful if any changes are made to the input folders (e.g. creating a new input file), which would otherwise require a restart of the GUI.

4.1.1.3 Input file *(menu)*

Select the simulation input file from the menu. The menu lists all the input files that are located in the "In" folder of the project folder. The menu is empty if no input files are found.

4.1.1.4 Edit / New *(button)*

Edit the selected input file or create a new input file if the input file menu is empty. The input file editor is opened after clicking the button (see section 4.2).

4.1.1.5 Top texture file *(menu)*

Select the texture file from the menu which will be used in the simulation to describe the top texture of the incoherent layer. The menu lists all the texture files that are located in the "Texture" folder of the project folder. A perfectly flat surface can be selected at the top of the menu.

4.1.1.6 Bottom texture file *(menu)*

Select the texture file from the menu which will be used in the simulation to describe the bottom texture of the incoherent layer. The menu lists all the texture files that are located in the "Texture" folder of the project folder. A perfectly flat surface can be selected at the top of the menu.

4.1.1.7 View *(button)*

View the selected top or bottom texture in a rendered 3D figure. The texture viewer is opened after clicking the button (see section 4.3).

4.1.1.8 Output file name *(text box)*

Enter the base name of the output files which will be created during the simulation. Six output files are created during each simulation, all of which have the same base name and different endings (see chapter 6 for a detailed description of the output files).

4.1.1.9 Browse *(button)*

Select the base name of the output files by browsing through the previously created output files. This is useful if trying to access the results of a previous simulation.

4.1.1.10 View 3D structure *(button)*

View the entire device structure in a rendered 3D figure, showing both the top and the bottom texture. The device structure viewer is opened after clicking the button (see section 4.4). The button is greyed out if no input file is selected.

4.1.1.11 View 3D rays *(button)*

View the ray propagation schematic in a rendered 3D figure. The schematic corresponds to the results of the simulation specified by the output name in the output file name text box (see section 4.1.1.8). The ray propagation viewer is opened after clicking the button (see section 4.5). The button is greyed out if no ray propagation figure file with the specified output name is found in the "Out" folder of the project folder (i.e. the simulation has not been completed yet, or the generation of ray propagation figure files has been disabled; see section 4.2.1.6).

4.1.1.12 View results *(button)*

Open the results viewer (see section 4.6). By default, the results of the simulation specified by the output name in the output file name text box (see section 4.1.1.8) are loaded into the simulation results list of the results viewer (see section 4.6).

4.1.1.13 Add to batch *(button)*

Add the current simulation (specified by the input file, the texture files, and the output file name) to the bottom of the current simulation batch list (see section 4.1.2). After adding a simulation to the batch, the project folder can no longer be changed (it is greyed out). Note that all the simulations in the batch must be run from within the same project folder.

4.1.1.14 RUN *(button)*

Execute a single CROWM simulation, using multiple processor cores. After executing the simulation, a Command Window is opened displaying the initialisation procedures, including the distribution of the load between the processor cores, followed by the current status of the simulation. The current status of the simulation is given by the index of the current ray in relation to the total number of rays that need to be simulated. After the simulation is completed, the window is closed.

(Please note that the current status of the simulation in the case of multi-core simulation is only useful for confirming that the simulation is running. It does not, however, reveal any information about the work done so far, since the order of the incident rays that are simulated is chosen randomly each time.)

4.1.1.15 RUN (single core) *(button)*

Execute a single CROWM simulation, using a single processor core. After executing the simulation, a Command Window is opened displaying the initialisation procedures, followed by the current status of the simulation. The current status of the simulation is given by the index of the current ray in relation to the total number of rays that need to be simulated, as shown in Figure 4.2. After the simulation is completed, the window is closed.

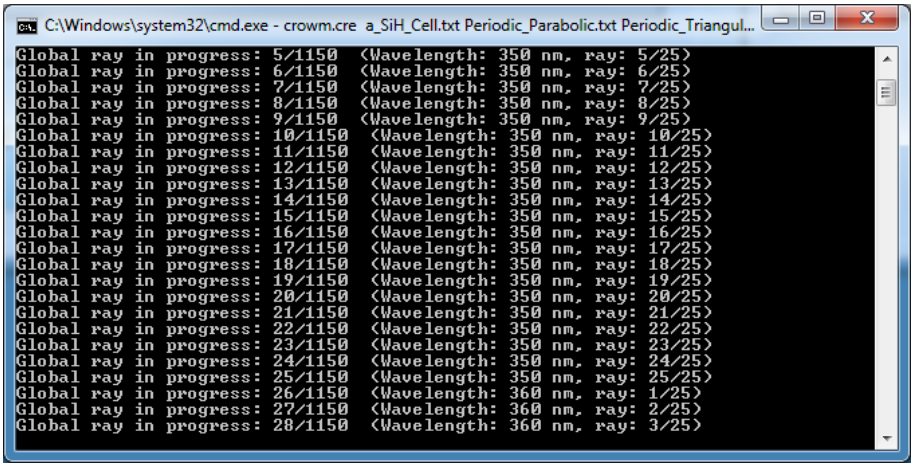

C:\Windows\system32\cmd.exe - crowm.cre a_SiH_Cell.txt Periodic_Parabolic.txt Periodic_Triangul...
Global ray in progress: 5/1150 (Wavelength: 350 nm, ray: 5/25)
Global ray in progress: 6/1150 (Wavelength: 350 nm, ray: 6/25)
Global ray in progress: 7/1150 (Wavelength: 350 nm, ray: 7/25)
Global ray in progress: 8/1150 (Wavelength: 350 nm, ray: 8/25)
Global ray in progress: 9/1150 (Wavelength: 350 nm, ray: 9/25)
Global ray in progress: 10/1150 (Wavelength: 350 nm, ray: 10/25)
Global ray in progress: 11/1150 (Wavelength: 350 nm, ray: 11/25)
Global ray in progress: 12/1150 (Wavelength: 350 nm, ray: 12/25)
Global ray in progress: 13/1150 (Wavelength: 350 nm, ray: 13/25)
Global ray in progress: 14/1150 (Wavelength: 350 nm, ray: 14/25)
Global ray in progress: 15/1150 (Wavelength: 350 nm, ray: 15/25)
Global ray in progress: 16/1150 (Wavelength: 350 nm, ray: 16/25)
Global ray in progress: 17/1150 (Wavelength: 350 nm, ray: 17/25)
Global ray in progress: 18/1150 (Wavelength: 350 nm, ray: 18/25)
Global ray in progress: 19/1150 (Wavelength: 350 nm, ray: 19/25)
Global ray in progress: 20/1150 (Wavelength: 350 nm, ray: 20/25)
Global ray in progress: 21/1150 (Wavelength: 350 nm, ray: 21/25)
Global ray in progress: 22/1150 (Wavelength: 350 nm, ray: 22/25)
Global ray in progress: 23/1150 (Wavelength: 350 nm, ray: 23/25)
Global ray in progress: 24/1150 (Wavelength: 350 nm, ray: 24/25)
Global ray in progress: 25/1150 (Wavelength: 350 nm, ray: 25/25)
Global ray in progress: 26/1150 (Wavelength: 360 nm, ray: 1/25)
Global ray in progress: 27/1150 (Wavelength: 360 nm, ray: 2/25)
Global ray in progress: 28/1150 (Wavelength: 360 nm, ray: 3/25)

Figure 4.2: Command Window during a simulation running on a single core.

4.1.2 Simulation batch setup

A simulation batch is a series of CROWM simulations which are prepared in advance to be run one after another when required (e.g. an overnight batch). Each line in the simulation batch list represents a single simulation. The simulations in the batch are executed in order, beginning from the top of the simulation batch list. A simulation batch can be saved, loaded, or rearranged as required by using the following controls.

4.1.2.1 Open *(button)*

Open a previously saved simulation batch.

4.1.2.2 Save *(button)*

Save the current simulation batch (as a ".bat" file). If a simulation batch is saved to the root of the project folder, it can be run as a standalone file without using the GUI (see chapter 3 for more information about how to run CROWM without using the GUI).

4.1.2.3 Remove *(button)*

Remove the selected (highlighted) simulation from the simulation batch list.

4.1.2.4 Clear *(button)*

Clear the entire simulation batch list.

4.1.2.5 Up *(button)*

Move the selected (highlighted) simulation up through the simulation batch list (it will be executed sooner).

4.1.2.6 Down *(button)*

Move the selected (highlighted) simulation down through the simulation batch list (it will be executed later).

4.1.2.7 RUN *(button)*

Execute all the CROWM simulations in the simulation batch list, using multiple processor cores, beginning from the top of the list. After running the simulation batch, a Command Window is opened displaying the initialisation procedures of each simulation. After all the simulations in the batch are completed, the window is closed.

4.2 Input file editor

The input file editor (shown in Figure 4.3) is used to setup all the parameters of the simulation which are stored in the simulation input file (see section 5.1). The full path and name of the input file which is currently being edited by the input file editor is displayed in the text box at the top of the input file editor window. To the right of the text box are buttons **Open** and **Save** which are used to open a different input file or to save the changes done to the current input file, respectively (the button **Save** turns red when any changes are done to the input parameters).

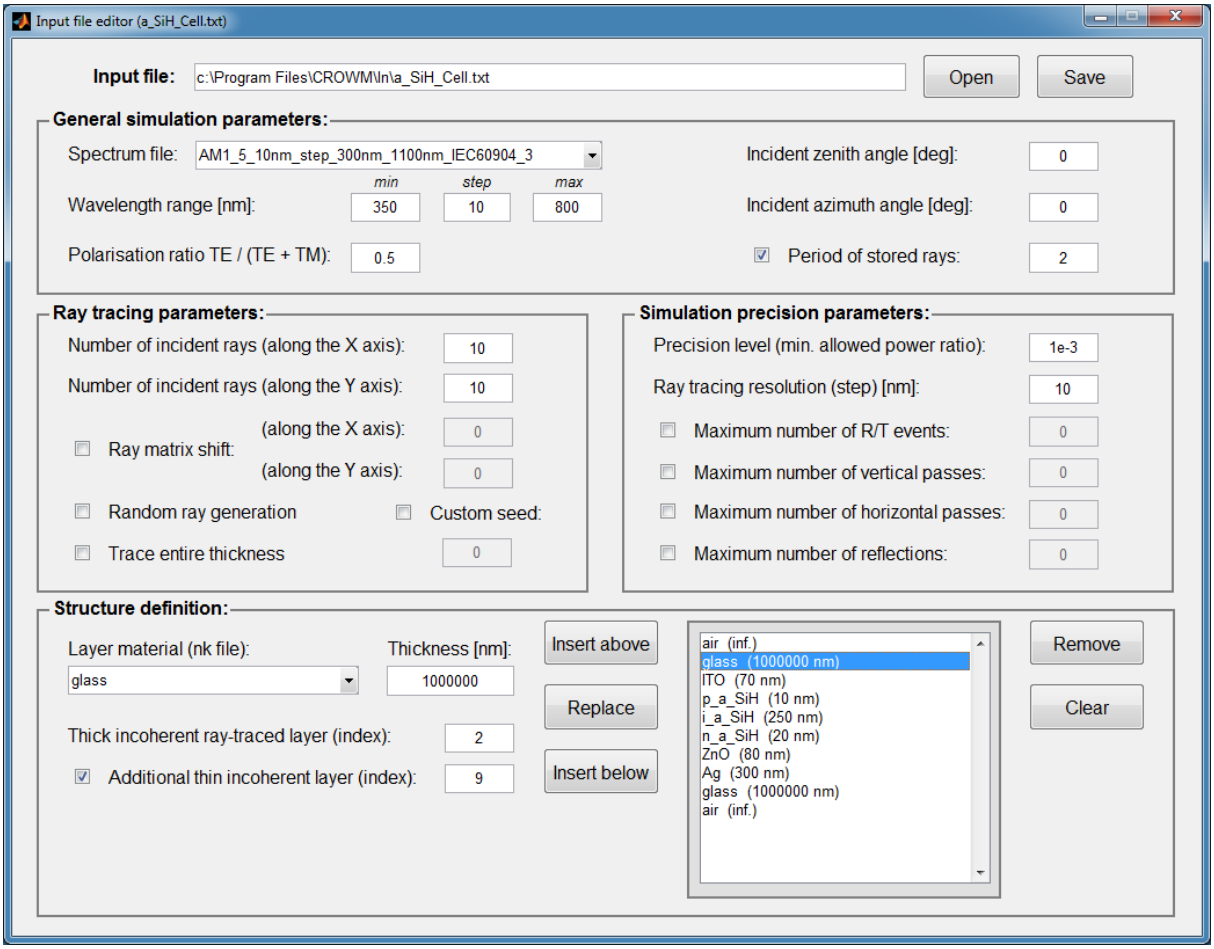


Figure 4.3: Input file editor.

Additionally, the input file editor is divided into four sections, each dealing with a specific aspect of the CROWM simulation. The functions of the controls contained within each of these sections are explained in the following.

4.2.1 General simulation parameters

4.2.1.1 Spectrum file *(menu)*

Select the spectrum file (see section 5.3) which will be used to calculate the short-circuit current densities (J_{sc}) from the reflectance, transmittance, and absorptance (RTA) results of the simulation. The spectrum file must be located in the "Spectrum" folder of the current project folder.

4.2.1.2 Wavelength range *(text box)*

Enter the wavelength range of the simulation: the minimum wavelength, the maximum wavelength, and the wavelength step (all in nanometres).

4.2.1.3 Polarisation ratio *(text box)*

Enter the power ratio between the TE-polarised and TM-polarised incident illumination. In the case of purely TE-polarised illumination, this setting should be set to 1. In the case of purely TM-polarised illumination, this setting should be set to 0. Global polarisation in CROWM is defined with respect to the horizontal *xy* plane – the electric field of the incident TE-polarisation is always parallel to the *xy* plane (the orientation of the coordinate system as assumed in CROWM is also shown in Figure 1.1).

4.2.1.4 Incident zenith angle *(text box)*

Enter the zenith angle (in degrees) of the incident illumination. Zenith angle values in the range of ±90° are allowed. In the case of perpendicular illumination (relative to the horizontal *xy* plane), the zenith angle should be set to 0°.

4.2.1.5 Incident azimuth angle *(text box)*

Enter the azimuth angle (in degrees) of the incident illumination. Azimuth angle values in the range of ±360° are allowed. The origin of the azimuth angle (0°) is along the *x* axis.

4.2.1.6 Period of stored rays *(check box & text box)*

The check box enables or disables creation of the ray propagation figure file (see section 6.4) during the simulation. Additionally, if creation of ray propagation figure file is enabled, the period of stored rays can be entered in the text box. If the period is set to 1, each ray will be stored; if it is set to 2, every second ray along each of the *x* and *y* axes will be stored, etc.

Please note that this setting does not alter the number of incident rays which are simulated (see section 4.2.2.1) but merely controls the size and density of the ray propagation figure file which is created during the simulation. If graphical representation of ray propagation is not required, this option should be disabled in order to save disk space.

4.2.2 Ray tracing parameters

4.2.2.1 Number of incident rays (along the X and Y axis) *(text box)*

In a typical CROWM simulation, the power of the incident illumination is evenly distributed among the incident rays which are evenly spaced along the *x* and/or *y* axis (also seen in Figure 4.6). This setting controls the number of rays to be generated along each axis.

Note that the numbers of rays generated along each of the axes do not need to be equal. Indeed, when performing CROWM simulations of device structures which only include 1D textures (i.e. the texture morphology changes only in one dimension), the user is advised to generate rays only along the dimension where the texture morphology changes, while keeping the number of rays along the other dimension at 1 (this in effect turns 3D simulations to 2D).

The total number of rays that need to be simulated is one of the critical parameters which significantly affect the speed of the simulation! Choosing the right number of rays is crucial for obtaining the most information in the least amount of time. For this decision, the textures of the thick incoherent layer present the most important factor. While indeed a single ray is enough to simulate a perfectly flat structure, the number of rays generally needs to be increased when the texture becomes more dynamic.

4.2.2.2 Ray matrix shift (along the X and Y axis) *(check box & text box)*

In a typical CROWM simulation, the incident rays are generated so that the origin of the first ray is located in the (0,0) point of the *xy* plane. By enabling ray matrix shift, the rays can be shifted along the *x* and/or *y* axis by the specified amount (0 – 1). This value presents the relative shift compared to the full distance between two adjacent rays along the chosen axis.

4.2.2.3 Random ray generation *(check box)*

This setting enables random generation of incident rays, which is an alternative to evenly spaced generation along the *x* and *y* axis. The total number of incident rays is still controlled by the product of the two values specified in the "Number of incident rays" setting, however the location of each ray is determined randomly.

4.2.2.4 Custom seed *(check box & text box)*

When using random ray generation, random number generation (*RNG*) functions need to be called. By default, the initial value (seed) of the *RNG* functions is determined automatically at the beginning of each simulation based on the system clock. However, by enabling this setting, the user has an option to use a custom number for the *RNG* seed. This way, if using the same seed in different simulations, the same random numbers will be generated and therefore the same results will be obtained.

4.2.2.5 Trace entire thickness *(check box)*

By default, in order to speed up simulations, incremental ray tracing between the lowest point of the top texture (minimum *z* value) and the highest point of the bottom texture (maximum *z* value) is not performed since the thick incoherent layer is perfectly uniform between these two points. Enabling this option overrides the default simulation method and forces ray tracing to be performed throughout the thick incoherent layer. Note that this setting results in significantly longer simulation times and should only be used when generation of full ray propagation figure file is required (see section 6.4).

4.2.3 Simulation precision parameters

4.2.3.1 Precision level (minimum allowed power ratio) *(text box)*

When a ray with the initial (incident) power P_0 is propagating through the device, its power P is diminished by numerous refractions, reflections, and absorption encountered on its path. If the power compared to the incident power drops below a certain limit specified by the precision level (e.g. $P/P_0 < 0.0001$ or 1e-4), the ray is ignored and the simulator proceeds to the next ray in the analysis. The power of the incident ray P_0 is defined as the total power given at the current wavelength in the spectrum file (see section 5.3) divided by the total number of incident rays: $P_0 = P_{spc}(\lambda)/N_{inc.rays}$.

An increased level of precision (lower power limit) leads to more accurate results, however, it also leads to dramatically longer simulation times! A careful balance thus must be achieved by selecting just the right precision level which will still result in accurate results but won't bog down the simulation to unreasonable durations. The setting of 0.001 (or 1e-3) will generally suffice in most cases.

4.2.3.2 Ray tracing resolution (step) *(text box)*

During a CROWM simulation, finite ray tracing steps are employed to incrementally trace the propagation of rays along their trajectories. A finer (lower) ray tracing resolution generally leads to more accurate results, however, it also leads to longer simulation times! Again, the most important factor in choosing the right ray tracing resolution is the nature of the textures – smaller and more dynamic surface features generally require shorter ray tracing steps (finer ray tracing resolution).

4.2.3.3 Maximum number of R/T events *(check box & text box)*

By means of this option, the user can limit the number of times that a single ray exhibits reflection or transmission events at an interface. If the specified maximum number of reflection/transmission events is reached, the ray is assumed to be absorbed in the current medium and the simulator proceeds to the next ray in the analysis.

4.2.3.4 Maximum number of vertical passes *(check box & text box)*

By means of this option, the user can limit the number of times that a single ray passes the entire thick incoherent layer from one texture to the other. If the specified maximum number of vertical passes is reached, the ray is assumed to be absorbed in the thick incoherent layer and the simulator proceeds to the next ray in the analysis.

4.2.3.5 Maximum number of horizontal passes *(check box & text box)*

By means of this option, the user can limit the number of times that a single ray passes horizontally from one lateral border to the other, before reaching the top or the bottom texture. If the specified maximum number of horizontal passes is reached, the ray is assumed to be absorbed in the thick incoherent layer and the simulator proceeds to the next ray in the analysis. This setting can also be used as a safety option to prevent a ray becoming infinitely trapped when propagating horizontally within a non-absorptive incoherent layer (i.e. its power is never reduced below the limit given by the precision level; see section 4.2.3.1). Normally, such a situation should not arise.

4.2.3.6 Maximum number of reflections *(check box & text box)*

By means of this option, the user can limit the number of times that a single ray reflects from the textures. If the specified maximum number of reflections is reached, the ray is assumed to be absorbed in the vicinity of the texture and the simulator proceeds to the next ray in the analysis. Note that in the case of a perfectly flat structure, the number of reflections is equal to the number of vertical passes. In the case of textured structures, however, the numbers are no longer the same (the ray may experience multiple reflections from a texture before being reflected to the other side of the thick incoherent layer).

4.2.4 Structure definition

4.2.4.1 Device structure *(panel)*

Device structure panel lists the refractive index files used for the layers and their thicknesses in order from top to bottom throughout the device.

4.2.4.2 Layer material *(menu)*

Select the refractive index file which will be used for the layer. Refractive index files are located in the "nk" folder of the project folder.

4.2.4.3 Thickness *(text box)*

Enter the thickness of the layer (in nanometres). Note that infinite thicknesses are assumed for the incident (top) medium and the medium in transmission (bottom).

4.2.4.4 Insert above *(button)*

Insert the current layer (combination of the refractive index file and layer thickness) above the selected (highlighted) layer in the device structure.

4.2.4.5 Replace *(button)*

Replace the selected (highlighted) layer in the device structure with the current layer (combination of the refractive index file and layer thickness).

4.2.4.6 Insert below *(button)*

Insert the current layer (combination of the refractive index file and layer thickness) below the selected (highlighted) layer in the device structure.

4.2.4.7 Remove *(button)*

Remove the selected (highlighted) layer in the device structure.

4.2.4.8 Clear *(button)*

Clear the entire device structure, reverting to the basic device structure comprised of a single glass layer surrounded by air.

4.2.4.9 Thick incoherent ray-traced layer (index) *(text box)*

Specify which layer in the device will be assumed as incoherent and analysed by means of ray tracing. Enter the index number of the target layer in the device structure, counting top-down through the list. Note that the incident (top) medium and the medium in transmission (bottom) are always assumed to be incoherent.

4.2.4.10 Additional thin incoherent layer (index) *(check box & text box)*

If enabled, specify a single additional thin layer which will be treated incoherently in the simulation within the framework of the transfer matrix formalism. Enter the index number of the target layer in the device structure, counting top-down through the list.

4.3 Texture viewer

Texture viewer is used for graphical representation of the selected texture. If the texture to be viewed contains a large number of data points (i.e. a large matrix in the texture file), the texture can be interpolated to a maximum of 200 x 200 points before being plotted in the texture viewer. An example parabolic texture displayed in the texture viewer is shown in Figure 4.4.

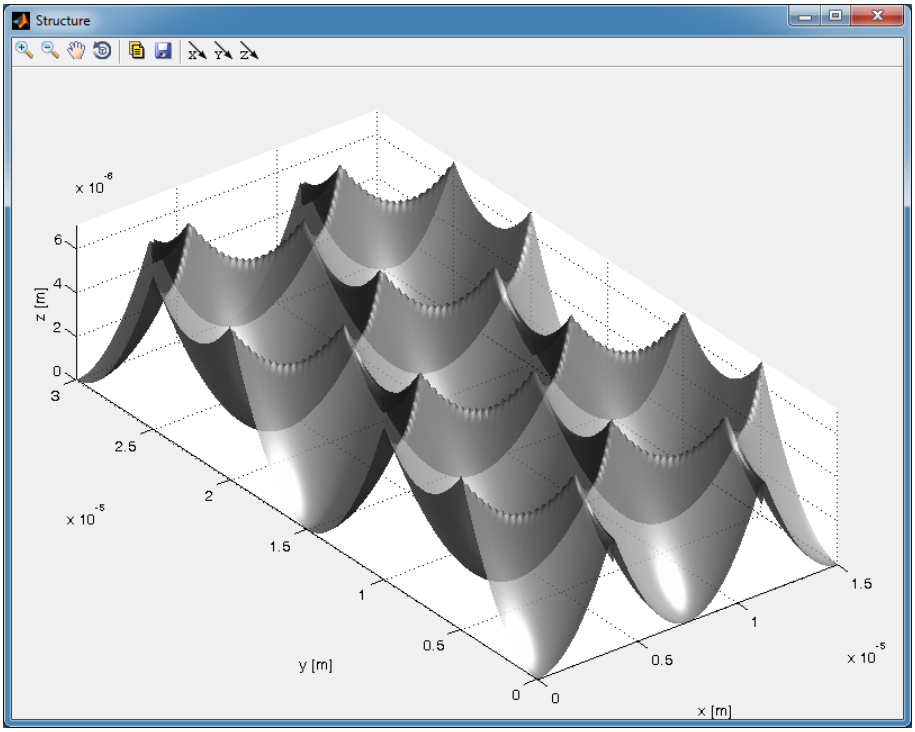


Figure 4.4: An example parabolic texture displayed in the texture viewer.

The following controls can be accessed from the toolbar in the top-left corner of the texture viewer (in order from left to right):

- **Zoom in**
- **Zoom out**
- **Pan**
- **Rotate**
- **Open texture file** (automatically opens the texture file in the default text editor)
- **Save figure** (save the figure as a bitmap file)
- **Set the point of view along the *x* axis** (view the texture in the *yz* plane)
- **Set the point of view along the *y* axis** (view the texture in the *xz* plane)
- **Set the point of view along the *z* axis** (view the texture in the *xy* plane)

4.4 Device structure viewer

Device structure viewer is used for graphical representation of the entire device structure, including both textures at the front and back side of the thick incoherent layer. If any of the textures contains a large number of data points (i.e. a large matrix in the texture file), the texture can be interpolated to a maximum of 200 x 200 points before being plotted in the device structure viewer. An example parabolic texture displayed in the texture viewer is shown in Figure 4.5.

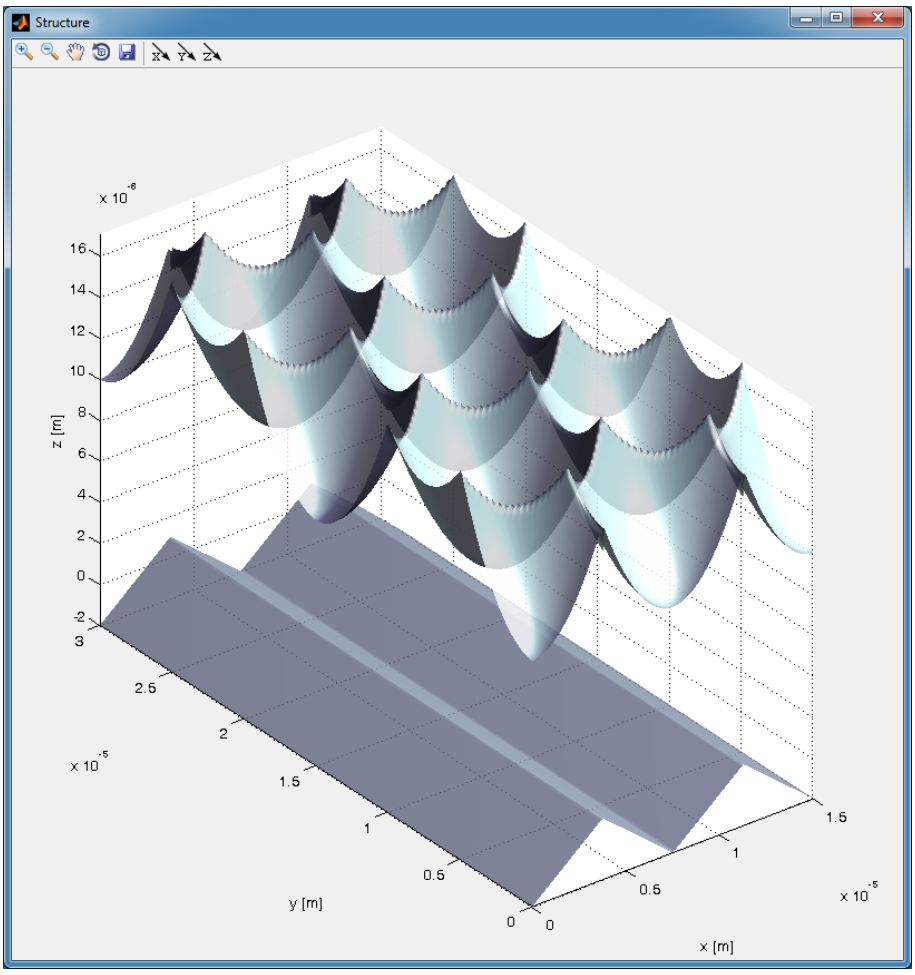


Figure 4.5: An example device structure displayed in the device structure viewer.

The device structure viewer features the same controls as the texture viewer (see section 4.3), except the **Open texture** file button.

4.5 Ray propagation viewer

Ray propagation viewer is used for graphical representation of ray propagation through the device structure. The viewer opens the ray propagation figure file which was created during the simulation (see section 6.4) and displays it in the ray propagation viewer window. Please note that creation of the ray propagation figure file is controlled by the "Period of stored rays" setting in the input file; see section 4.2.1.6. Propagation of rays through an example device structure is shown in Figure 4.6.

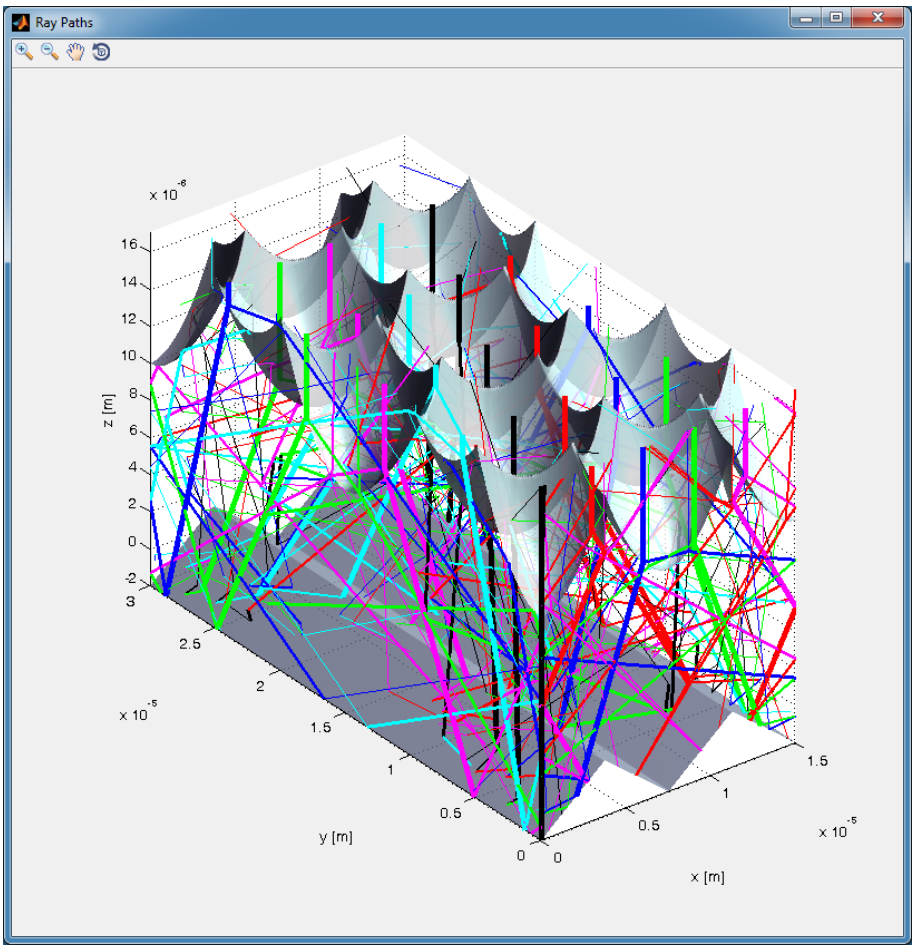


Figure 4.6: An example propagation of rays displayed in the ray propagation viewer.

The following controls can be accessed from the toolbar in the top-left corner of the ray propagation viewer:

- **Zoom in**
- **Zoom out**
- **Pan**
- **Rotate**

4.6 Results viewer

Results viewer (shown in Figure 4.7) is used for viewing the results of CROWM simulations. The basic controls can be accessed from the toolbar in the top-left corner:

- **Print** (prints the contents of the results viewer window)
- **Set plot boundaries** (used for editing the range of both plot axes)
- **Zoom in**
- **Zoom out**
- **Pan**
- **Legend** (toggles display of the legend)
- **Open RTA results** (automatically opens the "RTA" file in the default text editor)
- **Open Jsc results** (automatically opens the "Jsc" file in the default text editor)
- **Open log file** (automatically opens the "log" file in the default text editor)

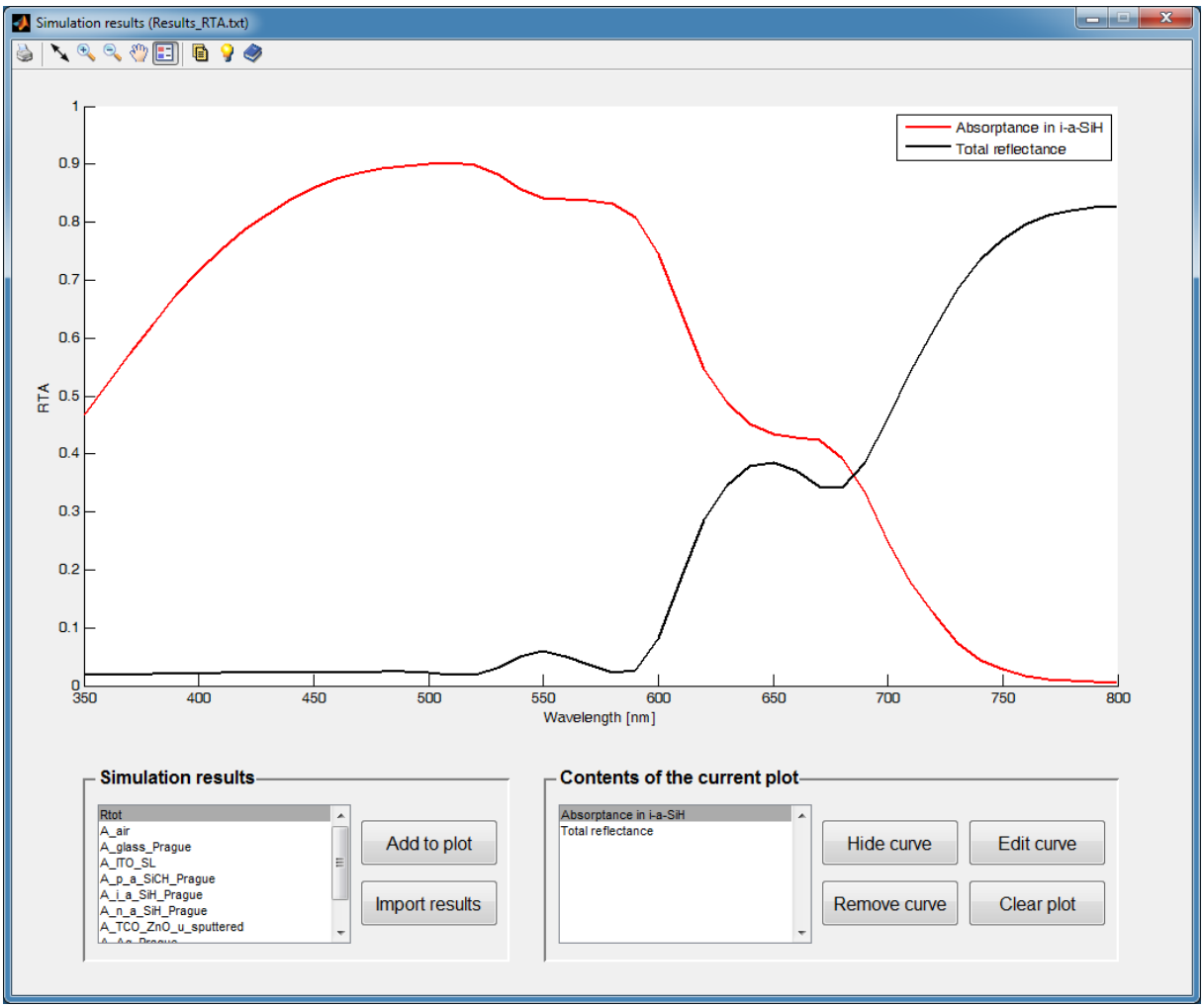


Figure 4.7: Results viewer.

The bottom part of the results viewer is divided into two sections. The first section is titled ***"Simulation results"*** and lists the reflectance, transmittance, and absorptance (RTA) results of the current simulation, whereas the second section titled ***"Contents of the current plot"*** lists the selected RTA results which are displayed in the current plot. The following controls can be accessed:

4.6.1 Add to plot *(button)*

Add the selected (highlighted) RTA simulation result to the current plot.

4.6.2 Import results *(button)*

Open RTA results from a previous simulation (see section 6.2).

4.6.3 Hide curve *(button)*

Hide the selected (highlighted) curve in the current plot.

4.6.4 Edit curve *(button)*

Edit the properties of the selected (highlighted) curve in the current plot. Curve colour and text displayed in the plot legend can be changed as desired.

4.6.5 Remove curve *(button)*

Remove the selected (highlighted) curve from the current plot.

4.6.6 Clear plot *(button)*

Clear all curves from the current plot.

5. DESCRIPTION OF THE INPUT FILES

The input files required for a CROWM simulation are located in four input folders: "In", "nk", "Spectrum", and "Texture". In the following, each of the input files is explained in detail. Please note that additional custom files can be created freely as long as the general structure of the files (as explained in the following sections) remains unchanged. Indeed it is often best and most convenient to use the existing files as "templates" and simply replace the relevant data and save the file under a new name.

5.1 Simulation input files (".txt", located in "In" folder)

The input files are located in the "In" folder of the project folder. An input file is a plain text file (".txt" extension) which lists all the important simulation parameters, such as the structure definition, the wavelength range etc. The name of the input file can be chosen freely as long as there are no spaces and no non-standard characters present in the text. For example, "Sim_0deg_10percent.txt" is a valid input file name, whereas "Sim 0deg 10%.txt" is not.

The contents of the input files can be most conveniently edited by means of the "Input file editor" window of the CROWM GUI (see section 4.2). Alternatively, they can also be edited in a standard text editor (e.g. Notepad). The input files are divided into four sections:

- **General simulation parameters**
- **Simulation precision parameters**
- **Ray-tracing parameters**
- **Structure definition**

IMPORTANT: The structure of the input files must not be changed in any way! All the spaces, titles, and empty lines must remain at their original position. The editable parameters in the first three sections must always begin on the 35th column.

In the following, the function of each of the settings in the input files will be explained. Where applicable, the user will be referred to section 4.2 where these settings have already been explained for the case of "Input file editor" of the CROWM GUI.

5.1.1 General simulation parameters

- **Spectrum file** – name of the spectrum file (see section 4.2.1.1)
- **TE/(TE+TM) ratio** – polarisation ratio (see section 4.2.1.3)
- **Wavelength start [nm]** – minimum wavelength of the simulation (see section 4.2.1.2)
- **Wavelength step [nm]** – wavelength step in the simulation (see section 4.2.1.2)
- **Wavelength stop [nm]** – maximum wavelength of the simulation (see section 4.2.1.2)
- **Incident zenith angle [deg]** – zenith angle of the incident rays (see section 4.2.1.4)
- **Incident azimuth angle [deg]** – azimuth angle of the incident rays (see section 4.2.1.5)
- **Period of plotted rays*** – period of rays in the ray propagation figure (see section 4.2.1.6)

5.1.2 Simulation precision parameters

- **Ray tracing precision level** – precision level (see section 4.2.3.1)
- **Ray tracing resolution [nm]** – ray tracing incremental step (see section 4.2.3.2)
- **Max No. of R/T events*** – max. number of refl. / trans. events (see section 4.2.3.3)
- **Max No. of vertical passes*** – max. number of vertical passes (see section 4.2.3.4)
- **Max No. of horizontal passes*** – max. number of horizontal passes (see section 4.2.3.5)
- **Max No. of reflections*** – max. number of reflections (see section 4.2.3.6)
- **Random number generator seed*** – custom seed for the *RNG* (see section 4.2.2.4)

5.1.3 Ray-tracing parameters

- **Number of incident rays (X axis)** – number of rays along the *x* axis (see section 4.2.2.1)
- **Number of incident rays (Y axis)** – number of rays along the *y* axis (see section 4.2.2.1)
- **Ray matrix shift factor (X axis)** – shift of the rays along the *x* axis (see section 4.2.2.2)
- **Ray matrix shift factor (Y axis)** – shift of the rays along the *y* axis (see section 4.2.2.2)
- **Random ray generation*** – randomly generated rays if set to 1 (see section 4.2.2.3)
- **Trace entire thickness*** – trace entire incoherent layer if set to 1 (see section 4.2.2.5)
- **Incoherent (ray-traced) layer** – index of the incoherent layer (see section 4.2.4.9)
- **Incoherent (thin) layer*** – additional thin incoherent layer (see section 4.2.4.10)

5.1.4 Structure definition

Structure definition consists of two lists – the first is the list of layers (refractive index files) as they appear top-down through the device, and the second is the list of the corresponding layer thicknesses (in nanometres). The thicknesses of the incident medium and the medium in transmission are set to 0.

All the above settings indicated with an asterisk (*) can be disabled when set to 0.

5.2 Complex refractive index files (".nk", located in "nk" folder)

The complex refractive index files are located in the "nk" folder of the project folder. They are used in the simulation to define the material of each layer in the analysed structure. A refractive index file is a plain text file with ".nk" extension. It consists of a simple textual header (ignored by the simulator) followed by three columns of data, as shown in Figure 5.1. The first column lists the wavelengths (in nanometres) at which the refractive index data is given. The second column lists the real refractive index of the material (n), and the third column lists the imaginary refractive index (or the extinction coefficient, k) for each wavelength. A fourth column listing the absorption coefficient (α) may also be present in the file, however it is ignored by the simulator.

```
Glass
wavelength        n           k
nm

300.000           1.530       1.671e-5
305.000           1.530       1.358e-5
310.000           1.530       1.104e-5
...               ...         ...
```

Figure 5.1: Data structure of an example refractive index file.

5.3 Spectrum files (".spc", located in "Spectrum" folder)

Spectrum files are located in the "Spectrum" folder of the project folder. They are used in the simulation to calculate the potential short-circuit current densities (J_{sc}) from the simulated absorptances in each of the layers (see section 6.3). A spectrum file is a plain text file with ".spc" extension. It consists of a simple textual header (ignored by the simulator) followed by two columns of data, as shown in Figure 5.2. The first column lists the wavelengths (in nanometres) at which the spectrum data is given. The second column lists the power density (in mW/cm^2) of each spectral component.

```
AM1.5 Spectrum IEC60904
wavelength        P
nm                mW/cm2

300.000           3.1430e-3
310.000           0.0653
320.000           0.2043
...               ...
```

Figure 5.2: Data structure of an example spectrum file.

5.4 Texture files (".txt", located in "Texture" folder)

Texture files are located in the "Texture" folder of the project folder. They are used in the simulation to define the texture morphology applied to the top and/or bottom surface of the thick incoherent layer. A texture file is a plain text file with ".txt" extension. It consists of a number of textual headers (ignored by the simulator) and two blocks of numerical data, as shown in Figure 5.3.

The first block (line) of numerical data contains the maximum *x* and *y* values (in micrometres) of the texture. In the example in Figure 5.3, the texture spans from 0 to 5 μm along the *x* axis, and from 0 to 10 μm along the *y* axis.

The second block of numerical data contains the matrix of *z* values (in micrometres), which represents the morphology of the texture as seen from above. The origin of the *xy* plane is assumed to be in the bottom left corner of the matrix. The *x* axis increases from left to right, and the *y* axis increases from bottom to top. The matrix of *z* values must have at least 4 elements in each dimension; however the number of elements in both dimensions does not need to be the same. The matrix of *z* values is interpolated to fit across the entire domain, as defined by the maximum *x* and *y* values. During this (linear) interpolation, the data points of the matrix are evenly distributed along each dimension.

Textures used in CROWM simulations need to be periodic in both dimensions! Therefore, each value at the right-most column of the *z* matrix should be the same as the value at the left-most column (in the same row), and each value at the top-most row of the *z* matrix should be the same as the value at the bottom-most row (in the same column). This can also be observed in the example texture file shown in Figure 5.3.

```
*******************************************************************************
*                      CROWM v2.6 SIMULATION - TEXTURE FILE                   *
*******************************************************************************

Example triangular texture.


LATERAL BORDERS OF THE SIMULATION DOMAIN (X_max Y_max - all dimensions in [um]):
-------------------------------------------------------------------------------

5 10


TEXTURE MORPHOLOGY (at least 4x4 matrix with Z values - all dimensions in [um]):
-------------------------------------------------------------------------------

0 1 2 3 2 1 0
0 1 2 3 2 1 0
0 1 2 3 2 1 0
0 1 2 3 2 1 0
```

Figure 5.3: Data structure of an example texture file.

5.5 Simulation batch files (".bat", located in root folder)

Simulation batch files are located in the root of the project folder. They are used for running a series of CROWM simulations. A batch file is a plain text file with ".bat" extension. It consists of a simple header followed by a list of CROWM simulation execution commands (see section 3.2).

6. DESCRIPTION OF THE OUTPUT FILES

The output files created during a CROWM simulation are stored in the "Out" folder. They all have the same basic name defined by the user (e.g. "Out_name"; see section 4.1.1.8), but different endings. There are five text files (".txt" extension) and (optionally) one Matlab figure file (".fig" extension) generated during each simulation:

- Out_name_Log.txt
- Out_name_RTA.txt
- Out_name_RTA_TE.txt
- Out_name_RTA_TM.txt
- Out_name_Jsc.txt
- Out_name_Rays.bmp

6.1 Log file ("_Log.txt")

The log file summarises the most important information about the CROWM simulation. It lists any potential errors that occurred during the execution (such as missing input data etc.), stores the names of the input and texture files for future reference, and saves the begin/end times of the simulation.

6.2 Reflectance, transmittance, absorptance files ("_RTA.txt")

Reflectance, transmittance, and absorptance files ("RTA files" in short) list the simulated wavelength-dependent total reflectance from the device, total transmittance through the device, and the absorptance within each layer of the device. The reflectance, transmittance, and absorptance are defined as the relative fractions of the incident light power that gets reflected, transmitted, or absorbed on its path through the structure, respectively:

$$R = P_{reflected}/P_{spc} \qquad T = P_{transmitted}/P_{spc} \qquad A = P_{absorbed}/P_{spc}$$

Three RTA files are constructed as the result of the simulation – one for the TE-polarized light, one for the TM-polarized light, and one for the total (combined) light (see section 4.2.1.3 for the definition of polarisation). The data structure of an example RTA file is presented in Figure 6.1.

Wavelength [nm]	Rtot	A_air	A_glass	...	Ttot
350.0000	0.0186	0.0003	0.0021	...	0.0000
360.0000	0.0190	0.0003	0.0016	...	0.0000
370.0000	0.0196	0.0004	0.0013	...	0.0000
...	...	...	...	...	...

Figure 6.1: Data structure of an example RTA file.

6.3 Short-circuit current density file ("_Jsc.txt")

The short-circuit current density file ("Jsc file" in short) lists the potential J_{sc} values calculated from the simulated absorptance spectra in each of the layers by applying the illumination spectrum specified by the input file (see section 5.3) according to the following equation:

$$J_{sc} = \frac{q}{hc} \int P_{spc}(\lambda) \cdot A_{layer}(\lambda) \cdot \lambda \cdot d\lambda$$

Please note that the J_{sc} values should be treated merely as indicators of short-circuit current gains/losses throughout the optoelectronic device. J_{sc} calculated for a TCO layer, for example, thus represents the optical losses due to the parasitic absorption in this layer, whereas J_{sc} calculated for an *i*-*a*-Si:H layer, on the other hand, represents the actual expected generated photocurrent (assuming that all the absorbed photons contribute to charge-carrier generation; i.e. $QE = A$).

6.4 Ray propagation figure file ("_Rays.fig")

The ray propagation figure file is a Matlab figure which contains the structure of the device (including both textures) superimposed by the trajectories of the rays as they are propagating through the device (also shown in Figure 4.6). Only the rays calculated for the last wavelength of the simulation are included. The thickness of each ray segment corresponds to the current power of the ray.

The generation of ray propagation figure file can be controlled by the "Period of stored rays" option in the input file (see section 4.2.1.6).